UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/706,231	11/12/2003	Seeta Hariharan	RPS920030097US1/2936P	6404

47052 7590 09/08/2008

IBM RP-RPS
SAWYER LAW GROUP LLP
2465 E. Bayshore Road, Suite No. 406
PALO ALTO, CA 94303

EXAMINER	
SHIN, KYUNG H	
ART UNIT	**PAPER NUMBER**
2143	

NOTIFICATION DATE	DELIVERY MODE
09/08/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

patent@sawyerlawgroup.com

Office Action Summary	Application No.	Applicant(s)	
	10/706,231	HARIHARAN ET AL.	
	Examiner	**Art Unit**	
	Kyung Hye Shin	2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 August 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *28-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *28-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Applicant's submission filed on 8/11/2008 has been entered.

2. This action is responding to application papers filed on **11-12-2003**. Claims **28 - 39** are pending. Claims **1 - 27** have been cancelled. Claim **28** is independent.

- The 112 1st rejection has been withdrawn. Claim 35 has been amended to specify the usage of an unsupported operation instead of an unsupported algorithm as per the specification.

Response to Arguments

3. Applicant's arguments filed 8/11/2008 have been fully considered but were not persuasive.

3.1 Applicant argues that, Hebert does not disclose a network processor; Hebert does not disclose multiple network processors; Hebert does not disclose a congestion and avoidance behavior application; Hebert does not disclose a host processor to operate the congestion and avoidance application. (Remarks Page 10-16)

A processor whose principal functions are concerned with network telecommunications is a network processor. Hebert discloses programming a processor that controls communications in a network interconnected environment. (Hebert col 3, l 61 - col 4, l 2: programmed to achieve necessary communications) Hebert discloses a processor that control communications in a network interconnected environment. Therefore, Hebert does disclose a network processor based on its definition.

> By definition, a network processor is defined as: “A programmable CPU chip that is optimized for networking and communications functions.” (http://www.pcmag.com/encyclopedia_term/0,2542,t=network+processor&i=47907,00.asp)

Yao discloses multiple network processors. (Yao Figure 1 (30)) Yao discloses a programmable interface for a network processor. (Yao para 027, ll 16-18: programmable commands input to the network processor) Yao discloses a network processor which is a programmable chip or a host processor. And, Yao discloses that one of the main functions of its network processors is to control and attempt to avoid congestion in a network environment.

Yao discloses a congestion control application as indicated in the claim limitation. Yao controls congestion for multiple network processors utilizing programmable XON/XOFF commands input to the network processor. (Yao para 027, ll 16-18: programmable commands) And, Yao discloses a network processor controlled by a

programmable interface as per claim limitation. (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control; para 006, ll 1-11; para 007, ll 29-45: XOFF message processed when high watermark reached; XON message processed when low watermark reached; congestion control method)

The specification discloses that the "generic APIs communicate with the congestion control application and the heterogeneous network processors". Yao communicates with a congestion control (programmable) program and the network processors.

3.2 Applicant argues that the referenced prior art does not disclose, a plurality of application programming interfaces API. (see Remarks Page 14)

Yao discloses a programmable interface for a network processor. (Yao para 027, ll 16-18: programmable commands) And, Yao discloses multiple network processors. Therefore, Yao discloses multiple programmable interfaces or APIs. And, Hebert discloses a generic (non-specific vendor) programming interface (API) for a processor that is used to control network communications.

According to the specification Applicant's invention is disclosed as:

(a): A **generic** (API) interface is used to communicate with and control a network processor. (disclosed by Hebert)

(b): A host processor controlling a congestion control application. (disclosed by Yao)

(c) Multiple network processors. (disclosed by Yao)

(d): The congestion control application controls multiple network processors utilizing an application programming interface (API interface). (disclosed by Yao)

(See specification Page 5:

The network also includes at least one host processor that utilizes at least one congestion control application. **The method and system comprise providing a plurality of generic application program interfaces (APIs). The generic APIs communicate with the congestion control application(s) and the heterogeneous network processors.** The generic APIs communicate with the congestion control application(s) in a network processor independent manner, but manage the congestion control and avoidance behavior of the heterogeneous network processors in a network processor specific manner. Thus, the generic APIs allow the congestion control application(s) to be network processor independent and to manage the congestion control and avoidance behavior of the heterogeneous network processors in the network processor specific manner.)

3.3 Applicant argues that the referenced prior art does not disclose, the Jarvis prior art referenced. (Remarks Page 19)

Jarvis is not used as a ground of rejection: (a) for the plurality of network processors claim limitation; (b) for the host processor claim limitation; or (c) for the application programming interface (API) interface claim limitation.

Claim Rejections - 35 USC § 103

The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. **Claims 28 - 39** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hebert et al.** (US Patent No. **6,134,618**) in view of **Yao et al.** (US PGPUB No. **20030126280**) and further in view of **Jarvis et al.** (US Patent No. **5,870,561**).

Regarding Claim 28, Hebert discloses a system for managing behavior of network

processors, the system comprising:

a) a first of the network processors (Hebert col 2, ll 63-67: telecommunications switch (network processor) being of a different model or version from a second of the plurality of network processors; (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic or non specific) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged)

b) a host processor (Hebert col 2, ll 56-63: host to switch interface for controlling telecommunications devices (network processors) application that manages network processors, the network processor is independent (Hebert col 3, ll 26-30; col 3, ll 46-55: generic API; no specific telecommunications switch (network processor); col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) such that the application need not have specific knowledge of a network processor's hardware, software, or firmware in order to manage the network processor's behavior; (Hebert col 3, ll 39-45: standardized interface for application development) and

c) a application programming interface (API) (Hebert col 1, ll 22-27: multiple telecommunications switches; col 2, ll 56-63: host to switch API interface (multiple APIs)), each of the plurality of APIs being usable by the host processor, to manage any of the plurality of network processors, none of the plurality of APIs being limited for use with a specific network processor model or version. (Hebert col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged; not limited to a specific telecommunications switch

or network processor)

Hebert does explicitly disclose a header containing congestion algorithm information. However, Jarvis discloses information such as an algorithm transferred within an API interface used to control a congestion control program:

d) the multi-word header usable for congestion control, wherein the multi-word header is comprised of a plurality of words where a first part of the multi-word header consisting of a first plurality of words is common to the congestion control application and a second part of the multi-word header consisting of a second plurality of words is common to a plurality of congestion algorithms. (Jarvis col 5, ll 1-10: application programming interface (API); make requests and receives recommendation (information) concerning controlling congestion (generation of network traffic))

Hebert-Jarvis does not explicitly disclose multiple network processors and a congestion control application.

However, Yao discloses for a): a plurality of network processors controlling network traffic; for b): a congestion control application; the plurality of network processors for c): congestion control application. (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control; para 006, ll 1-11; para 007, ll 29-45: XOFF message processed when high watermark reached; XON message processed when low watermark reached; congestion control method); and a plurality of APIs. (Yao para 027, ll 16-18: programmable commands (programmable interface

(API)); para 005, ll 1-4: multiple network processors)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a congestion control application as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14: *" … The XON/XOFF flow control scheme prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. Other advantages will be apparent to one of ordinary skill in the pertinent arts. … "*)

Regarding Claim 29, Hebert discloses the system of claim 28, wherein the application of the host processor need not be modified in order to add a new network processor model or version to the system. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged; col 3, ll 31-35: additional features to be added without implementing additional context specific signaling) Hebert does not explicitly disclose a congestion control application. However, Yao discloses a congestion control application. (Yao para 006, ll 1-11; para 007, ll 29-45: XOFF message processed when high watermark reached; XON message processed when low watermark reached; congestion control method)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a congestion control application as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 30, Hebert discloses the system of claim 28, wherein the application of the host processor uses an API. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose a congestion control application, a plurality of network processors, and a location in network processor where behavior is to be managed. However, Yao discloses wherein a congestion control application (Yao para 006, ll 1-11; para 007, ll 29-45: congest control method), a plurality of network processors (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control), and a location in network processor where behavior is to be managed. (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control), and identify a location in network processors behavior is to be managed. (Yao para 006, ll 1-11: congestion controlled at port (XON/XOFF port control mechanism))

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a congestion control application, multiple network processors, and location to

control congestion as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 31, Hebert discloses the system of claim 30. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose that one network processor includes an ingress side and an egress side. However, Yao discloses wherein the identified location in the one network processor includes an ingress side and an egress side. (Yao Figure 1 ((45); (50)); para 004, ll 1-5: network processor; input port (ingress side), output port (egress side))

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a network processor that includes an ingress side and an egress side as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 32, Hebert discloses the system of claim 31. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol

specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose that the ingress side of the identified location in the one network processor includes a plurality of ports, a plurality of receive queues, and a plurality of receive flows. However, Yao discloses wherein the ingress side of the identified location in the one network processor includes a plurality of ports, a plurality of receive queues, and a plurality of receive flows. (Yao Figure 1 ((45); (50)); para 004, ll 1-5: multiple ports; para 016, ll 1-8: receive (ingress) queues or flows)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a network processor that includes multiple ports and receive queues as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 33, Hebert discloses the system of claim 31. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose a plurality of scheduler flows, a plurality of scheduler queues, a plurality of transmit queues, and a plurality of ports. However, Yao discloses wherein the egress side of the identified location in the one network processor includes a plurality of scheduler flows, a plurality of scheduler queues (Yao para 016, ll 8-12: scheduler arbiter (schedule flows)), a plurality of transmit queues (Yao para 016, ll 1-8: virtual output;

transmit queues), and a plurality of ports (Yao Figure 1 ((45); (50)); para 004, ll 1-5: multiple ports).

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a plurality of scheduler flows, a plurality of scheduler queues, a plurality of transmit queues, and a plurality of ports as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 34, Hebert discloses the system of claim 30, wherein the application of the host processor uses the plurality of APIs. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged)

Hebert does not explicitly disclose congestion control to apply at the identified location in the one network processor. However, Yao discloses wherein congestion controls to apply at the identified location in the one network processor. (Yao para 006, ll 1-11: XON/XOFF flow control algorithm; identified location is port)

It would have been obvious to one of ordinary skill in the art to modify Hebert to apply congestion control at the identified location in the one network processor as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents

problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Hebert-Yao does not explicitly disclose selecting a congestion control algorithm. However, Jarvis discloses wherein to select a congestion control algorithm. (Jarvis col 3, ll 15-27: select congestion control algorithm)

It would have been obvious to one of ordinary skill in the art to modify Hebert-Yao to select a congestion control algorithm as taught by Jarvis. One of ordinary skill in the art would have been motivated to employ the teachings of Jarvis in order not overload certain network links, particularly during periods of peak usage by restricting the use of selected network links based on the type and priority of the network traffic. (Jarvis col. 2, lines 20-27: " *… Moreover, the large volume of network traffic generated by application programs often overloads certain network links, particularly during periods of peak usage. This type of overloading may interfere with significant network traffic, such as file server and print server operations. Consequently, some network administrators would prefer to restrict the use of selected network links based on the type and priority of the network traffic serviced over the links. …* ")

Regarding Claim 35, Hebert discloses the system of claim 34 wherein the plurality of APIs by the network processor. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) However, Hebert discloses wherein returning a null behavior to the congestion control application of the host processor when an operation

in not supported or when a function is not implemented by the one network processor. (Hebert col 7, ll 49-53: predetermined sequence of functions invoked upon the occurrence of a particular event such as an unsupported operation; col 8, ll 43-45: determination whether a valid event for a normal state; event could be determination of an invalid operation)

Hebert does not explicitly disclose a congestion control application. However, Yao discloses a congestion control application. (Yao para 006, ll 1-11; para007, ll 29-45: XOFF message processed when high watermark reached; XON message processed when low watermark reached; congestion control method)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a congestion control application as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 36, Hebert discloses the system of claim 28 wherein the usage of an API. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose a configure API, an update API, an enable API, a disable API, and a list API.

However, Jarvis discloses:

a) the configure API being usable by the congestion control application of the host processor to configure the congestion and avoidance behavior of any of the plurality of network processors, (Jarvis col 5, ll 63-66: set programmable threshold limits for congestion control (configure))

b) the update API being usable by the congestion control application of the host processor to update the congestion and avoidance behavior of any of the plurality of network processors, (Jarvis col 5, ll 63-66: change (update)congestion control information (policies))

c) the enable API being usable by the congestion control application of the host processor to enable congestion control algorithms for any of the plurality of network processors; the disable API being usable by the congestion control application of the host processor to disable congestion control algorithms for any of the plurality of network processors, (Jarvis col 4, ll 57-61: enable/disable policy (congestion control capability)) and

d) the list API being usable by the congestion control application of the host processor to view congestion and avoidance information concerning any of the plurality of network processors. (Jarvis col 4, ll 65-67: view (list) congestion control policies))

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a configure API, an update API, an enable API, a disable API, and a list API as taught by Jarvis. One of ordinary skill in the art would have been motivated to employ the teachings of Jarvis in order not overload certain network links, particularly during

periods of peak usage by restricting the use of selected network links based on the type and priority of the network traffic. (Jarvis col. 2, lines 20-27)

Regarding Claim 37, Hebert discloses the system of claim 28, wherein the network processor resides in a switch or a router. (Hebert col 2, ll 63-67: telecommunications switch controlled by generic API) Yao does not explicitly disclose a plurality of network processors. However, Yao discloses wherein a plurality of network processors. (Yao para 005, ll 1-4: multiple network processors providing flow control (congestion))

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a plurality of network processors as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Regarding Claim 38, Hebert discloses the system of claim 28. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose a plurality of network processors to control network traffic. However, Yao discloses wherein a plurality of network processors to control network traffic. (Yao Figure 1 (30); para 005, ll 1-4: multiple network processors providing flow control)

It would have been obvious to one of ordinary skill in the art to modify Hebert to

use a plurality of network processors to control network traffic as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Hebert-Yao does not explicitly disclose a plurality of networks. However, Jarvis discloses a plurality of networks. (Jarvis col 4, ll 35-49: WAN (wide area network); multiple interconnected LANs)

It would have been obvious to one of ordinary skill in the art to modify Hebert to use a plurality of networks as taught by Jarvis. One of ordinary skill in the art would have been motivated to employ the teachings of Jarvis in order not overload certain network links, particularly during periods of peak usage by restricting the use of selected network links based on the type and priority of the network traffic. (Jarvis col. 2, lines 20-27)

Regarding Claim 39, Hebert discloses the system of claim 28. (Hebert col 3, ll 26-30; col 3, ll 46-55: universal (generic) API; col 3, l 61 - col 4, l 2: supports multiple protocol specific state machines; host to switch interface unchanged) Hebert does not explicitly disclose how packets are handled when the network processor is unable to process every packet during a particular interval. However, Yao discloses wherein congestion and avoidance behavior of a network processor includes how packets are handled by the network processor when the network processor is unable to process every packet

during a particular interval. (Yao para 030, ll 10-14: prior art invention substantially eliminates HOL block; HOL blocking occurs if prior art cannot successfully handle congestion problem or process every packet)

It would have been obvious to one of ordinary skill in the art to modify Hebert for how packets are handled when the network processor is unable to process every packet as taught by Yao. One of ordinary skill in the art would have been motivated to employ the teachings of Yao in order to use a XON/XOFF flow control scheme that prevents problems caused by HOL blocking such as increased system latency, unintentionally dropped packets, and time-out problems. (Yao para 030, ll 10-14)

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kyung Hye Shin whose telephone number is (571) 272-3920. The examiner can normally be reached on 9:30 am - 6 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tonia L. Dollinger can be reached on (571) 272 - 4170. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kyung Hye Shin
Examiner
Art Unit 2143

KHS
September 1, 2008

/Tonia LM Dollinger/

Supervisory Patent Examiner, Art Unit 2143